**The Store App**

**Software Requirements Specification**

**Version #1**

**Team Number: 10**

**Project Manager: Zach Devore**

**Mentor: Subbu Devarajan**

**Team Members:** **Simon Chaudhary, Mohamed Ibensilalen, Emma Ghostling, Brendon Seaton, Zach Devore**

**Revisions**

| **Version** | **Primary**  **Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| 1.0 | Peter Scott, Khaled El Khatib, Shelby Sanchez-Herrera, Wesley Smith, Zachary Jones, Kaleb White, Simon Chaudhary | Completed Sections 1-8, including all subsections, with data for Sections 1-4 coming from Milestone 1. | 10/16/2022 |
| 1.1 | Peter Scott, Khaled El Khatib, Shelby Sanchez-Herrera, Wesley Smith, Zachary Jones, Kaleb White, Simon Chaudhary | Reorganized section 2.1  Sections 5-8 expanded upon and reworded | 10/23/2022 |
| 2.0 | Peter Scott, Khaled El Khatib, Shelby Sanchez-Herrera, Wesley Smith, Zachary Jones, Kaleb White, Simon Chaudhary | Added sections 10.1-10.8 | 10/30/2022 |
| 2.1 | Peter Scott, Khaled El Khatib, Shelby Sanchez-Herrera, Wesley Smith, Zachary Jones, Kaleb White, Simon Chaudhary | Revised sections 10.2, 10.3, 10.4, and 10.8 | 11/6/2022 |

**Review History**

| **Reviewer** | **Version Reviewed** | **Date** |
| --- | --- | --- |
| Patrick Haberern | Version 1.0 | 10/17/2022 |
| Patrick  Haberern | Version 2.0 | 11/2/2022 |

**Table of Contents (version 1)**

1. Introduction
   1. Project Objectives
   2. Project Scope
   3. Project Overview
2. General Description
   1. Project Features / Functions
   2. User Stories
   3. Use Cases
3. Team Collaboration and Documentation Tools
4. Project Management Plan
5. Business Requirements
6. User Requirements
7. Functional Requirements
8. Non-Functional Requirements
9. Implementation (Performance) Requirements
10. High-level Design
    1. Security
    2. Hardware
    3. User Interface
    4. Architecture
    5. Database
    6. Top-level Classes
    7. Data Flow and States
    8. Reports
    9. Training
11. **Introduction**
    1. Project Objectives: To help a store increase its revenue with an application for an online store front.
    2. Project Scope: This project is designed to assist customers by allowing them to order items online without constricting their options or complicating their experience. The application designed in this project will grant the customer access to a store’s complete inventory.
    3. Project Overview: The application for the online store front will give a customer access to the store’s entire inventory available to purchase from anywhere. The home page will display any deals of the day and top selling products. Deals may be selected based on previous purchases, holidays, or general sales. There will also be a search bar so users can look for any specific item they are looking for. A product’s page will display the price of the item, including money saved if the item is on sale, a description of the item (including nutritional details for food), and a space at the bottom to display other products commonly purchased with the item, as well as alternatives to the item.

The app will allow customers to login to a profile or create a new profile. There will be a page that will display the customer's information and allow them to edit the information.

A shopping cart will keep track of the items being purchased and will allow customers to remove items they no longer want.

Customers will also be able to return items if they fit the criteria of the return policy.

Return Policy: Items delivered in a damaged state will be able to be returned. If a customer is unsatisfied with their order, they can return the item and receive store credit equal to the item’s price. All returns must be made within 15 days (about 2 weeks) of receiving the product

1. **General Description**
   1. Project Features / Functions
      1. A home page displaying top selling products and daily deals
      2. Ability to create a personal account using a unique username and password. Accounts can also be linked to phone numbers or emails for notifications or promotional material.
      3. Login functionality for returning customers
      4. An Account page displaying account details, order history, favorite orders (set by customer)
      5. Deals selected based on holidays, general sales, or previous purchases
      6. A search bar to help find a specific item
      7. A Product page for each item in stock
      8. Product page displaying item price, including money saved if item is on sale, a description of the item (including nutritional details for food), and space to display other products commonly purchased with the item or alternatives to the item.
      9. A Shopping cart screen keeping track of each item being purchased
      10. Ability to remove an item from the shopping cart
      11. Ability to adjust the quantity of an item from the shopping cart
      12. A return system allowing for the return of items that fit the criteria.
   2. User Stories
      1. As a customer, I need to see the entire inventory, so I can make a choice and add it to my shopping cart. >.
      2. As a customer, I must register (if new) and sign in to see the history of previous orders, to get discounts on new purchases for being a loyal customer and edit my orders.
      3. As a customer, I should be allowed to cancel a recent order if it was made within 24 hours for a full refund.
      4. As a customer, I should be allowed to return my item if I am not satisfied or if the item was damaged on arrival within 15 days (about 2 weeks).
      5. As a customer, I should have recommendations based on previous purchases (if applicable) or upcoming holidays.
      6. As a customer, I should be able to view the availability of assorted products.
      7. As a customer, I should be able to set notifications for sales or restock on items.
   3. Use Case
      1. **Login to or Register account**

**ID:** 1

**Brief Description:** Attempt to log into your account and if you do not have an account then it prompts you to sign-up for an account.

**Primary Actors:** A user on the store website

**Secondary Actors:** none

**Preconditions:**

* The user has access to the website

**Main Flow:**

* + - * 1. On any page the user selects login/register
        2. The option to log in is displayed
        3. The user then enters their username/email and password
        4. The system then checks to see if an account with that info exists
        5. If the account does not exist, they are prompted to try again or sign-up
        6. If they try again and the account does exist, they are logged in
        7. The user is taken back to the page they were on prior to

**Postconditions:**

* The user is signed into their account

**Alternative Flows:**

* An account does not exist, and they click sign-up
* The user is taken to the sign-up page
* The user enters their account information
* An account is created for them
* They are taken to their account page or back to the page they were on prior to
  + 1. **Buying items**

**ID:** 2

**Brief Description:** The user browses the catalog of items and can purchase the items they want if it is in stock.

**Primary Actors:** A user on the store website

**Secondary Actors:** none

**Preconditions:**

* The user has access to the website
* The item is in stock

**Main Flow:**

* + - * 1. On one of the pages with items for sale the user clicks an item they want to look at
        2. The item is displayed as well as information about the item and the stock currently available
        3. If the user clicks add to cart proceed to step 4
        4. The user is prompted to select the quantity they want of the item
        5. If the user selects back instead return to step 2
        6. If the user selects a quantity and adds it to cart the item is added to their cart
        7. If the user clicks check-out the checkout page is opened
        8. The user has the option to put in their purchase information or remove items from the cart
        9. The user is prompted to log-in to their account or sign-up for an account to save their order in their orders list
        10. The user can then put in their address and ship the order using several options for speed

**Postconditions:**

* The user can purchase their item

**Alternative Flows:**

* The user decides they do not want the items in their cart
* The options are to empty the cart and return to the store page or change the quantity of an item in their cart
  + 1. **Managing Orders**

**ID:** 3

**Brief Description:** If the user has an account and has bought something in the past then they can view their prior orders and cancel or return them.

**Primary Actors:** A user on the store website.

**Secondary Actors:** none

**Preconditions:**

* The user has an account
* The user is logged in
* The user has made a purchase

**Main Flow:**

* + - * 1. The user clicks the account page
        2. The user clicks the orders page
        3. The system brings up all past orders they have made while logged in
        4. If the user has bought something within 24 hours proceed to step 5
        5. The option to cancel is on the order
        6. If the user clicks the cancel button their order is cancelled
        7. If the user has had a package arrive at their address within 15 days (about 2 weeks) proceed to step 7
        8. The option to return their package is on the order

**Postconditions:**

* The user was able to cancel an order made within 24 hours
* The user was able to return an order within 15 days of arrival

**Alternative Flows:**

* The user does not have any orders that can be returned or canceled
* The user can see all their past orders with date of purchase

1. **Team Collaboration and Documentation Tools**

Microsoft Teams, Google Docs.

1. **Project Management Plan** (optional)

This project incorporates the main values of agile software development. We prioritize individuals and interactions, customer collaboration, and we review often and adjust where change is needed.

1. **Business Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| BR1 | Notify customers when products are in back in stock or on sale. | S |
| BR2 | The program must be user-friendly. | M |
| BR3 | Must include all inventories- and provide accurate inventory within 60 Seconds | M |

1. **User Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| UR1 | Search filters to help customers find the right product for them. | S |
| UR2 | Production information, including Product Number, and price and quantities | M |
| UR3 | Customers could order as a guest | C |

1. **Functional Requirements**

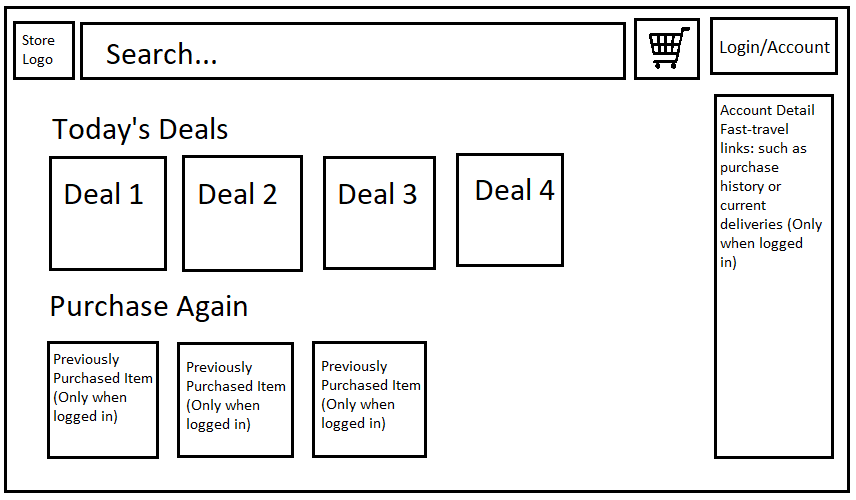
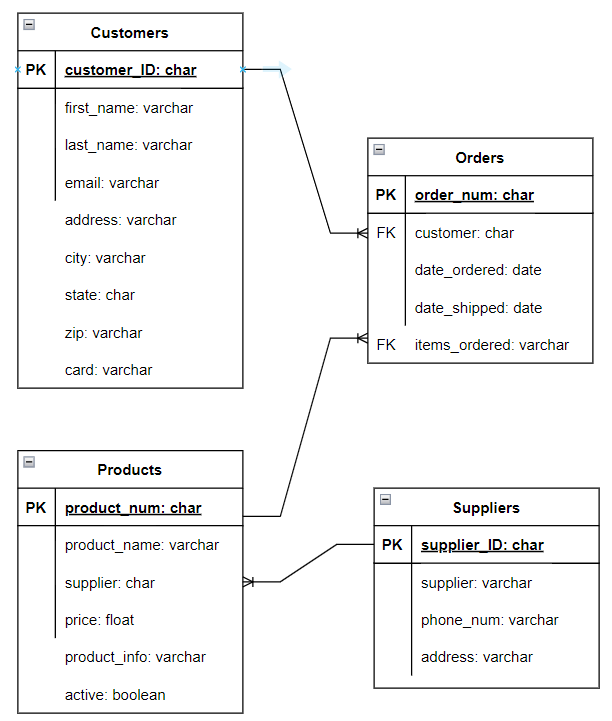
| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| FR1 | Add and remove items from customers shopping cart. | M |
| FR2 | Create a mobile app or mobile friendly website. | M |
| FR3 | Allowing customers to share a link to a product’s page via msg or in social media. | M |

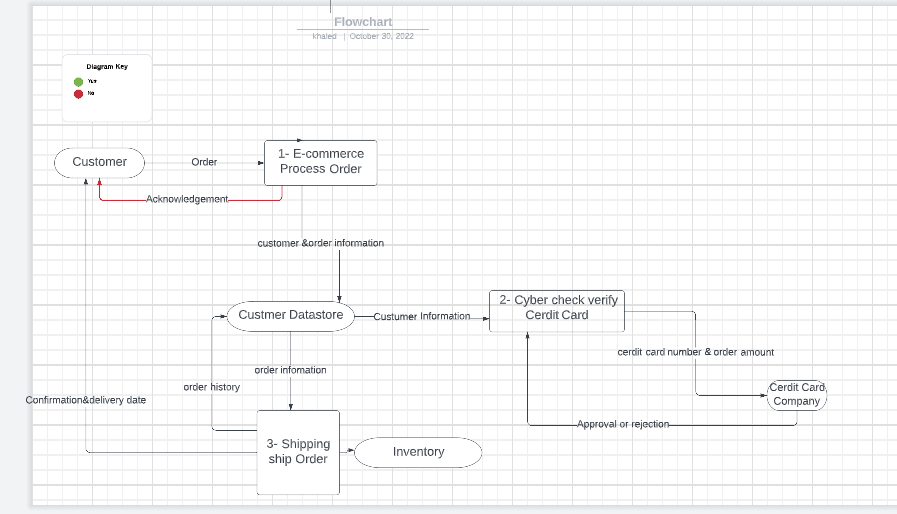
1. **Non-Functional Requirements**

| Requirement ID | Requirement Description | MOSCOW |
| --- | --- | --- |
| NFR1 | Customer’s username must be unique when creating a new account | M |
| NFR2 | Send email to customers on their birthday | C |
| NFR3 | Keep customers' information safe - Credit card information and other data | M |

1. **Implementation (Performance) Requirements**

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement Description | MOSCOW |
| IR1 | User friendly design: Clean and concise UI design that makes shopping easy to navigate and helps users get where they need to go with suggestions for them based on previous pages visited. | M |
| IR2 | 24-hour customer services. | M |
| IR3 | Third party integrations: allowing other companies to add their features like: collecting information of customers, adding tools that filter and sort product while searching for specific item. | M |

1. **High-level Design**
   1. **Security.**
      1. Use SHA 256 hash for security
      2. Use asymmetric encryption for user information as well as payment information
   2. **Hardware (Required)**
      1. The servers and database we will be using are Platform as a Service (PaaS) cloud-based service hosted by AWS. This limits our need for expensive hardware to get a network established.
      2. The only computer hardware requirement would be for the developers to design the website as well as pushing updates. A computer with decent processing power as well as a strong modem with low latency and fiber-optic internet would be beneficial to roll out changes and monitor the website in real time. If needed server-side processing could be an option.
      3. There should be a single server on site to hold a backup for our website in case of emergency but with the network hosted on AWS we will just be renting server space from them.
   3. **User Interface** 
      1. Each item on the User Interface will be labelled or use clear imagery to symbolize its purpose. Items will also have clear space, so the site is not cluttered.
      2. Items will be laid out in a grid-like fashion with general items such as the search bar or account info near the top, while more personalized content will be further down.
      3. 
   4. **Use Architecture (Required)**
      1. Using Three-Tier Architecture: Three layers of architecture which include, Presentation layer (client) which will be designed using HTML and CSS, Application layer (business logic) which will have the framework created using React for frontend and NodeJS for backend, Database layer (database) our database can be hosted on the AWS servers we will rent space on and will be hosted using MySQL. Operates as a separate module on a separate server.
   5. **Database** 
      1. Database Form see below
      2. Main tables and who modified the tables
      3. Entity Relationship Diagram
   6. **Top-level Classes** 
      1. Top level classes include the Customer class and Product class (see below)
      2. Class DiagramDiagram

         Description automatically generated
   7. **Data Flow and States (Required)**
      1. Data Flow Diagram
   8. **Reports (Required)**
      1. When an order is processed, an e-mail is sent to the customer. The e-mail will contain the order number, a list of products with prices, the total of the order, and the date and time the order was submitted.
      2. Once the order is shipped another e-mail is sent to the customer. The e-mail will have the order number, the shipment tracking number, the date, and time it was shipped, an estimated date of arrival, and any product that could not be fulfilled with an updated total.
      3. A customer can request these e-mails to be re-sent to them from the order history page.
   9. **Training** 
      1. **Our software is designed to be user-friendly, intuitive, and simple. No strict training regimen is necessary. We could offer demonstrations and/or a contact us page allowing users to ask questions regarding our software.**